# OUTAGE MITIGATION FOR GNSS/MEMS NAVIGATION USING NEURAL NETWORKS

*J.-R. De Boer*(1), *V. Calmettes*(2), *J.-Y. Tourneret*(1), *and B. Lesot*(3)

(1)Université de Toulouse
ENSEEIHT, IRIT
2 rue Charles Camichel B.P. 7122
31071 Toulouse Cedex 7, France
www.irit.fr

(2)Université de Toulouse
ISAE, DEOS, TESA
10 av. Edouard Belin B.P. 54032
31055 Toulouse Cedex 4, France
www.isae.fr

(3)Thales Avionics
25 rue Jules Védrines
26000 Valence, France
www.thalesgroup.com

email: Jean-Remi.De.Boer@enseeiht.fr, Vincent.Calmettes@isae.fr, Jean-Yves.Tourneret@enseeiht.fr and Bertrand.Lesot@fr.thalesgroup.com

**ABSTRACT**

*Inertial navigation systems (INS) and global navigation satellite systems (GNSS) are often combined to ensure high accuracy navigation. The last generation of inertial measurement units referred to as micro-electro-mechanical systems (MEMS) might also be used in a lot of applications thanks to their relatively low cost. However, the information given by the MEMS is less accurate than with classical INS. In particular, the performance of an integrated GNSS/MEMS navigation systems decreases drastically during GNSS outages. This paper studies a neural network based procedure that allows one to compensate this performance loss.*

## 1. INTRODUCTION

Vehicle navigation systems often include an inertial measurement unit (IMU) to bridge GNSS satellite outages (signal blockage) or GNSS signal degradation (jamming, multipath). Efficient inertial navigation systems (INS) include accurate but high cost IMUs. Recently, a new generation of inertial sensors called micro-electro-mechanical systems (MEMS) has become available at relatively low cost. These sensors might be used in many mass-market applications, e.g., for land vehicles. However, the performance of these systems is largely dependent upon the quality of inertial sensors, especially during GNSS outages. A possible solution for compensating navigation errors is to use learning strategies resulting from the theory of neural networks (NN) or fuzzy logic. These strategies were used successfully for hybrid GNSS/INS navigation in land vehicular applications [1, 2]. In these previous studies, an offline training of navigation errors was achieved and an adaptation of the initial system was performed online. This paper shows that NN can also be used for online training of GNSS/MEMS navigation errors.

The paper is organized as follows. Section 2 recalls the main elements of GNSS/MEMS navigation with a particular interest on the hybridization procedure. Section 3 presents the NN based strategy that will be studied in this paper. The application of this strategy to GNSS/MEMS navigation is investigated in Section 4. Simulation results shown in Section 5 allows one to appreciate the performance of the proposed navigation algorithm.

## 2. NAVIGATION PRINCIPLES

As described in [3, p. 3], navigation can be defined as the accurate determination of a position and a velocity relative to a known reference. Navigation can be achieved using many different techniques leading to dead reckoning navigation [4], celestial navigation [5], inertial navigation [3, Sec. 6], [6, Sec. 9] or satellite based navigation [3, Sec. 5], [6, Sec. 2.3]. This paper focuses on inertial and satellite navigation systems as well as their hybridization that will be described in the next sections.

### 2.1 Inertial navigation

Largely described in the literature [3, Sec. 6], [6, Sec. 2.2], this type of navigation is based on the integration and projection of accelerations given by accelerometers to provide velocity and then position. The fundamental idea comes from Newton's second law of motion. Of course, It will need also the estimation of the orientation of the sensor which can be expressed by the Euler angles computed thanks to angular rates given by rate gyro.

Accelerations (resp. angular rates) are not directly observable from the IMU's outputs, i.e., accelerometer outputs (resp. gyroscope outputs), which are available. These measures are corrupted by different noises that will produce navigation errors. As described in [3, Sec. 6.4], the navigation errors can arise from different sources including instrumentation errors, computational errors, alignment errors and environment errors. Neglecting computational errors, the navigation errors will depend upon the three remaining sources that are due to accelerometers and gyroscopes in the case of INS. A very general model for MEMS accelerometers and gyroscopes can be written

$$S = \Phi(\Gamma, \Omega, \theta), \tag{1}$$

$$\text{where} \quad \begin{cases} S & \text{is the MEMS output,} \\ \Gamma & \text{is the true acceleration,} \\ \Omega & \text{is the true angular rate,} \\ \theta & \text{contains the intrinsic MEMS parameters,} \\ \Phi & \text{is the MEMS nonlinearity function.} \end{cases}$$

The interested reader is invited to consult [7], [6, p. 214-219] for more details about model 1. A classical inertial platform is represented in Fig. 1 where the complexity of the inertial navigation problem can be easily observed. This complexity results from coordinate frame transformations,

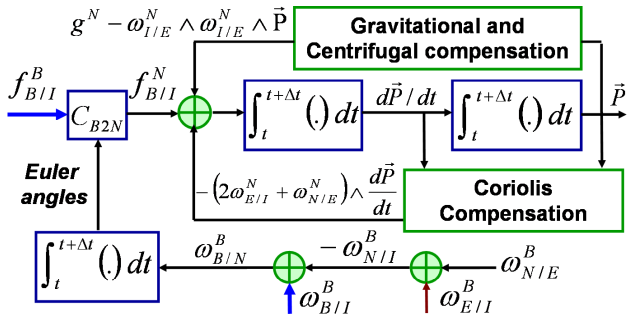


Figure 1: Inertial platform defined in the navigation frame.

different physical phenomenon compensation, . . . In this figure, $f^B_{B/I}$ and $\omega^B_{B/I}$ are the accelerometer and gyroscope outputs of the body frame with respect to the inertial frame coordinatized in the body frame, $f^N_{B/I}$ is the specific force of the body frame with respect to the inertial frame coordinatized in the navigation frame obtained from the transformation matrix $C_{B2N}$ and $\omega^B_{B/N}$ is the angular rate of the body frame with respect to the navigation frame coordinatized in the body frame obtained after compensation of both inertial rotation rate of the earth $\omega^B_{E/I}$ and transport rate $\omega^B_{N/E}$ of the navigation frame relative to the earth (see [3, p. 187-197] for more details). It is important to note that the different measurements resulting from the inertial platform are not contaminated by any external source of noise. However, it is well known that navigation based on inertial sensors leads to errors that are unbounded over time.

### 2.2 Satellite navigation

GNSS is the standard generic term for satellite-based radionavigation systems. These systems include the global positioning system (GPS), the Russian global navigation satellite system (GLONASS), the European Galileo or the Chinese Beidou (Compass) that are currently under investigation. This paper focuses on GPS (that is the only fully operational system) whose principle is simple but whose implementation is more tedious. The main idea of GPS is to measure the elapsed time between the emission of radio waveforms from satellites with known positions and their arrivals to the mobile to be located. Knowing radio wave velocity, the range between the satellites and the mobile can be determined, as well as the mobile position by triangulation. However, the range is corrupted by many noise sources. The observed measurements (usually referred to as pseudo-ranges) can be mathematically described as follows

$$\rho^i = R^i + c(\Delta_{\tau_R} - \Delta_{\tau_{Si}}) + c\Delta_{\tau_a} + \omega^i, \tag{2}$$

where
$$\begin{cases} \rho^i & \text{is the pseudo-range between the receiver} \\ & \text{and the } i\text{th satellite } S^i, \\ R^i & \text{is the true range between the receiver} \\ & \text{and the } i\text{th satellite } S^i, \\ c & \text{is the light velocity,} \\ \Delta_{\tau_R} & \text{is the receiver clock error,} \\ \Delta_{\tau_{Si}} & \text{is the } i\text{th satellite clock error,} \\ \Delta_{\tau_a} & \text{is a delay due to atmospheric crossing,} \\ \omega^i & \text{is a residual error.} \end{cases}$$

The pseudo-ranges can be contaminated by different noise sources including signal jamming, signal outages and multipath. However, it is well known that estimation errors resulting from the resolution of (2) are bounded.

### 2.3 Hybrid navigation

The complementary of GNSS and INS systems has been used efficiently in many applications: short-term position errors of INS are relatively small but degrade without remaining bounded whereas GNSS errors do not degrade with time but are more important during a short time interval. Many different GNSS/INS integration architectures have been proposed in the literature [3, Sec. 7.2 and 7.3], [6, Sec. 10]. This paper focuses on a tightly coupled architecture for GPS aided INS integration depicted in Fig. 2. The associated state model can be written as

$$\begin{cases} \dot{X}_t = A_t X_t + B_t v_t, \\ Y_t = h_t(X_t) + w_t, \end{cases} \tag{3}$$

where
$$\begin{cases} X_t & \text{is the state vector,} \\ Y_t & \text{is the observation vector,} \\ A_t & \text{is the state transition matrix,} \\ B_t & \text{is the state noise matrix,} \\ h_t & \text{is the observation function,} \\ v_t & \text{is the process noise,} \\ w_t & \text{is the observation noise.} \end{cases}$$

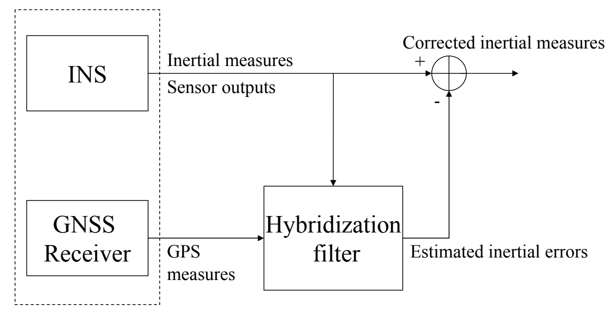


Figure 2: Hybridization architecture.

The GNSS/INS state vector is classically defined as $X_t = (\delta p_t, \delta v_t, \delta\rho_t, b_{s,t}, b_t, \dot{b}_t)$ where

$$\begin{cases} \delta p_t \text{ are the INS position errors,} \\ \delta v_t \text{ are the INS velocity errors,} \\ \delta \rho_t \text{ are the INS attitude errors,} \\ b_{s,t} = (b_{a,t}, b_{g,t}) \text{ are the accelerometer and gyroscope biases,} \\ (b_t, \dot{b}_t) \text{ are the GPS receiver clock bias and drift.} \end{cases}$$

The estimation of $X_t$ from the state model (3) can be achieved by many different filtering strategies based on the extended Kalman filter (EKF), the unscented Kalman filter or particle filtering. This study concentrates on the classical EKF for simplicity reasons.

## 3. NEURAL NETWORKS

The human brain is composed of a great number of neurons that are organized in a huge network allowing learning and memorizing. An artificial NN is a collection of elementary neurons (defined in 3.1) trying to model the human brain.

Training the NN consists of estimating its parameters by minimizing an appropriate cost function. In this study, we will use a typical cost function defined as the squared error between the NN output and a known desired output associated to an element of the learning set. After training has been performed, the NN can be used in its "generalization mode", i.e., the NN output is computed for data vectors that do not belong to the training set. More details regarding training and generalization are provided below.

### 3.1 Elementary cell: the Neuron

The output of an elementary neuron is defined as a weighted combination of its inputs that pass through a nonlinear activation function $\varphi$ (as shown in Fig. 3). The activation function used in this paper is the hyperbolic tangent yielding the following input/output relationship:

$$\mathrm{OUT} = \tanh\left(\sum_{j=1}^{n} w_j \, \mathrm{IN}_j\right). \tag{4}$$

The NN weights are the intrinsic parameters that have to be estimated during the "training mode". This estimation can be conducted using different optimization algorithms based on gradient descent, Newton's methods or mixed strategies (e.g., the Levenberg-Marquardt algorithm that combines advantages from gradient descent and Newton's methods).

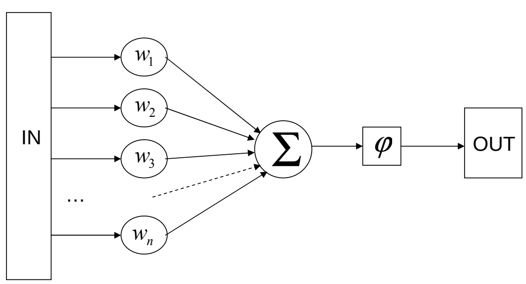


Figure 3: Generic elementary neuron.

### 3.2 Neural network structure

An NN is a collection of elementary cells organized in one or several layers as depicted in Fig. 4. The NN structure is defined by the number of layers and the number of neurons in each layer. It is well known that any continuous multivariate function (after appropriate normalization yielding outputs in $[-1,+1]$) can be approximated by a single hidden layer NN [8]. However, the NN structure is generally determined empirically (by cross validation). An NN with two hidden layers with ten neurons on each layer has provided good results for the current application.

### 3.3 Neural network training

As described in [9], the main property of the NN is its ability to learn from its environment and to improve its performance through learning. This learning can be conducted with a teacher (supervised learning) or without teacher (unsupervised learning)(the teacher has a certain knowledge of the environment thanks to input-output examples belonging to the learning set). This paper focuses on supervised learning that is achieved using the popular back-propagation algorithm [10], [9, Sec. 4.3]. The learning rule used in this paper is very classical and summarized below:

1. Initialization,
2. Presentations of training samples,
3. Forward computation,
4. Backward computation,
5. Iteration.

Training can be stopped when the cost function is lower than an appropriate threshold, or when the maximum authorized number of epochs (number of presentations of the learning set) is reached. Fig. 5 shows an example of cost function (blue line) and threshold (black line) versus the number of epochs presented to the NN. This example corresponds to one experiment detailed in section 5. This figure allows one to adjust the number of epochs required to obtain a required precision for training.

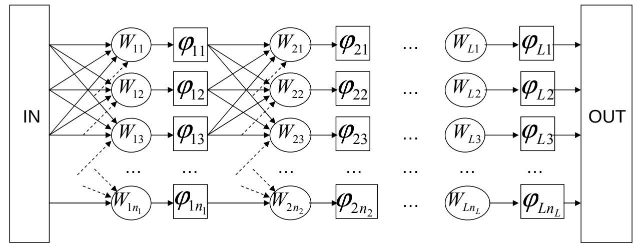


Figure 4: Neural network structure.

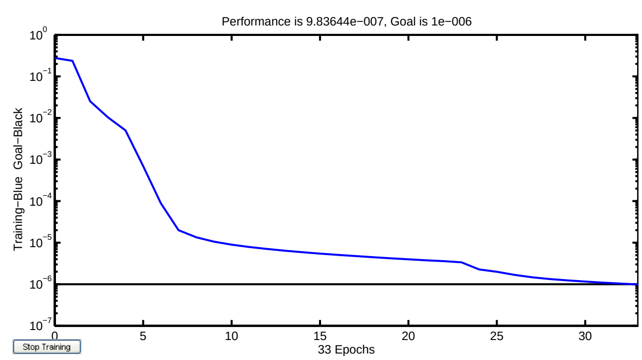


Figure 5: Convergence example for one experiment presented in this paper.

## 4. NN-AIDED GNSS/MEMS INTEGRATION

As explained before, hybrid GNSS/MEMS navigation should be interested in many practical applications. Indeed, because of their low cost, MEMS sensors might be used in mass market applications. However, the performance of MEMS in stand alone mode, i.e., during GNSS outages, is not satisfactory. The idea developed in this paper is to learn the behavior of the GNSS/MEMS system thanks to an NN whose learning is performed during the time intervals without outages. This knowledge is then used to improve navigation performance during outages.

The main difference between the proposed methodology and previous studies is the learning strategy. Indeed, the training was achieved offline in [1] and [2] with a large variety of input/output data in order to cover most of the anticipated various dynamics and motion scenarios during real navigations. The system was then adapted online using error corrections resulting from training. Here and contrary to the analysis conducted in [1] and [2], MEMS parameters vary from one run to another. Thus, the proposed algorithm does not use any offline training. Instead, error corrections are learnt online using the past of the vehicle motion. More precisely, the NN weights are adjusted from intput/output samples collected during time intervals that are not subjected to outages. We want to use the difference between inertial and hybrid navigation to compensate errors during outages. Consequently, GNSS outages will be simulated online in training mode and will allow us to determine the difference between inertial and hybrid navigation. A similar strategy was proposed in [11] with a different NN architecture. More precisely, the NN input vector used in this study contains the following features

$$\begin{cases} \Delta_t: & \text{elapsed time from beginning of outage,} \\ S_a: & \text{accelerometer outputs,} \\ S_g: & \text{gyroscope outputs,} \\ \Delta_{p_{MEMS}}: & \text{distance from beginning of outage,} \\ v_{MEMS}: & \text{inertial estimations of velocities,} \\ \rho_{MEMS}: & \text{inertial estimations of attitudes.} \end{cases}$$

The NN outputs allowing us to compensate inertial errors (in terms of position) are defined by the differences between hybrid and inertial position estimates, denoted as $p_{GNSS/MEMS} - p_{MEMS}$. The learning set used in this study is composed of simulated 30s outages (we have considered that an outage cannot exceed a duration of 30s). The resulted training architecture is shown in Fig. 6.

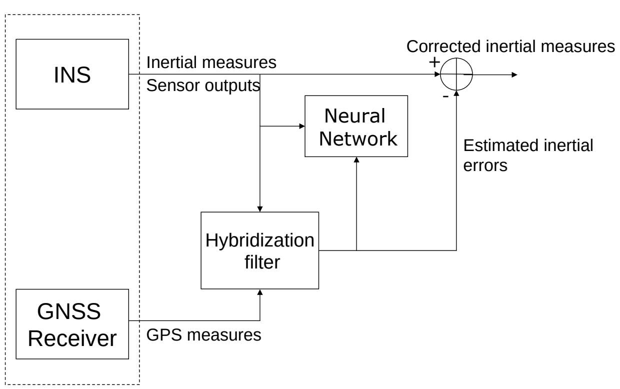


Figure 6: Neural network training architecture.

When a real GNSS outage is encountered, the NN works in its generalization mode to compensate inertial position errors as depicted in Fig. 7.

## 5. SIMULATION RESULTS

The results presented in this paper are obtained by calculating the position mean absolute error (MAE) defined by (5) where $p$ is the real position and $\hat{p}$ is the hybrid position estimates with or without NN corrections. This MAE is com-

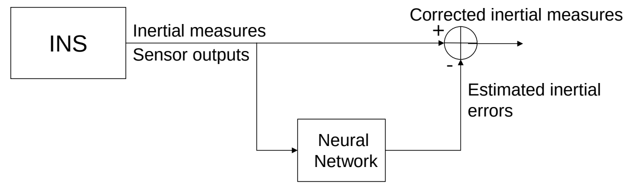


Figure 7: Neural network generalization architecture.

puted after a hundred of Monte Carlo runs as follows

$$MAE = \frac{1}{n}\sum_{j=1}^{n} \|p - \hat{p}\|, \qquad (5)$$

where $n = 100$. Three different experiments with 30s GNSS outages will be discussed. The first experiment considers a mobile with fixed position. A constant velocity trajectory will be studied in the second experiment. The last experiment is characterized by a real trajectory in ISAE campus Supaéro.

### 5.1 Experiment 1: fixed position

The first experiment considers a 30s outage in the interval $[1350s, 1380s]$. The associated MAE are shown in Fig. 8 for GNSS/MEMS (thick blue line) and NN-aided GNSS/MEMS (thin green line). The advantage of using the NN for compensating the navigation errors is clearly shown on this example. Indeed, the MAE after a 30s outage is about 400m without NN whereas it is approximately 100m when NN is used for error compensation. Note that the estimated position is better without NN corrections during the first 5s of the outage showing that the NN has some difficulties to learn small differences between hybrid and inertial position estimates.

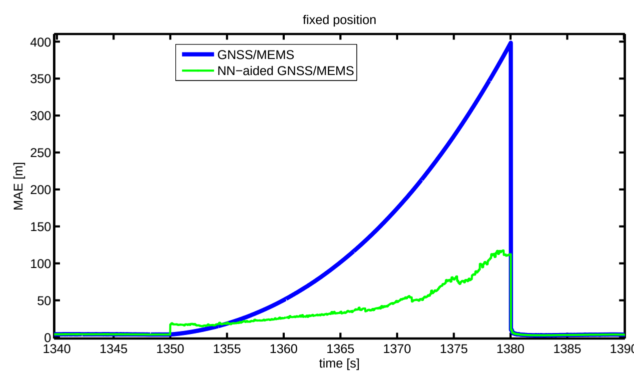


Figure 8: Position mean absolute error (fixed position).

### 5.2 Experiment 2: constant velocity

The second experiment considers a constant velocity trajectory. The Fig. 9 shows the same type of results than in the previous experiment. The NN-aided integration system provides better navigation performance during the outage when compared to the navigation without NN corrections (except during the first 5s of the outage as previously). After a 30s outage, the position accuracy is about 500m without NN corrections whereas it is close to 100m in the NN-aided case.

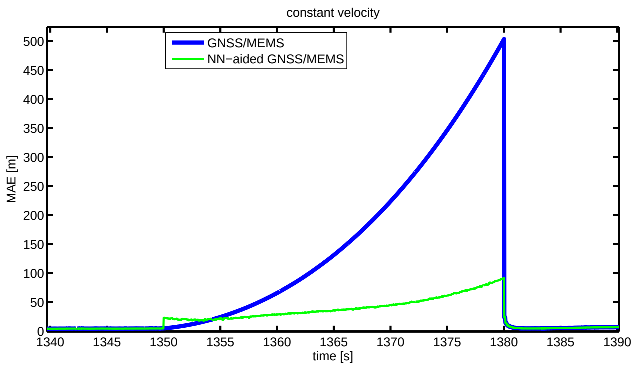


Figure 9: Position mean absolute error (constant velocity).

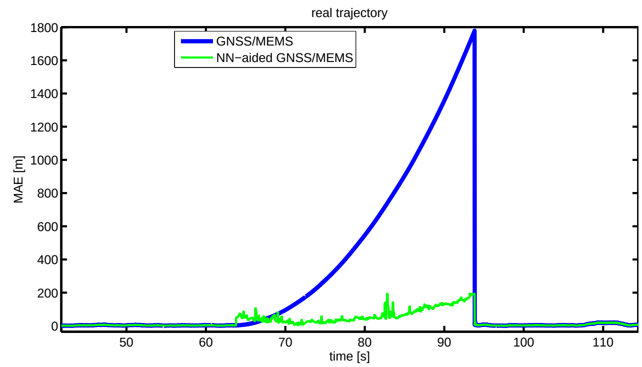


Figure 11: Position absolute error (real trajectory).

### 5.3 Experiment 3: real trajectory

The last experiment studies a real trajectory in Toulouse, France (ISAE campus Supaéro) depicted in Fig. 10. An 30s outage (green line in Fig. 10) is simulated during this trajectory (red line in Fig. 10). Fig. 11 shows position absolute error for a particular run in presence and absence of NN error corrections. NN-aided GNSS/MEMS integration clearly provides more accurate position estimates than GNSS/MEMS without NN corrections.

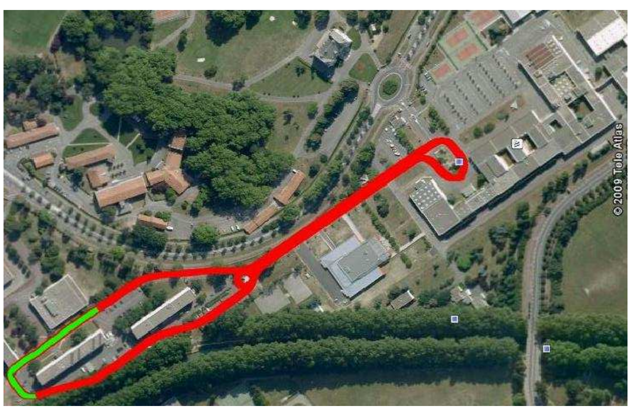


Figure 10: Reference trajectory in Toulouse, France.

## 6. CONCLUSION

Current progress in MEMS technology has made possible the use of inertial sensors for low cost mass market applications. However, the performance of a GNSS/MEMS system is closely related to the IMU quality, especially during critical navigation scenarios. These scenarios include urban canyon characterized by partial satellite outages, poor geometric dilution of precision (GDOP), and total satellite outages. This paper proposed an approach to improve the quality of MEMS based navigation systems during these critical scenarios by using an appropriate NN to compensate navigation errors.

The strategy implemented in this study uses an NN to learn on-line the system behavior during time intervals that do not contain satellite outages. This learning is then used advantageously in the presence of outages. Simulation results illustrated the performance of the proposed navigation strategy during GNSS outages for different state models (mobile with fixed position, constant velocity, real trajectory).

Progresses in MEMS technology should continue in the future decade and might lead to high accuracy navigation systems. Further investigations include the reduction of training time by simulating superposed outages and the validation of the proposed NN-aided GNSS/MEMS integration on real data. Our future works also contain the development of new MEMS calibration procedure based on nonparametric system identification.

## ACKNOWLEDGEMENT

The authors would like to thank DGA and Thales Avionics for supporting this work.

## REFERENCES

[1] K.-W. Chiang, "INS/GPS integration using neural network for land vehicular navigation applications," Ph.D. dissertation, University of Calgary, 2004.

[2] W. Abdel-Hamid, A. Noureldin, and N. El-Sheimy, "Adaptative fuzzy prediction of low-cost inertial-based positionning errors," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 3, pp. 519–529, June 2007.

[3] J. Farrell and M. Barth, *The Global Positioning System & Inertial Navigation*. New York: McGraw-Hill, 1998.

[4] A. Ayliffe, "The development of airborne dead reckoning. part I: Before 1940 finding the wind," *The Journal of Navigation*, vol. 54, pp. 223–233, May 2001.

[5] O. S. Brown, *One Day Celestial Navigation for Offshore Sailing*, 2nd ed. Washington: Washington Book Distributors, 2002.

[6] M. Grewal, L. Weill, and A. Andrews, *Global Positioning System, Inertial Navigation, and Integration*, 2nd ed. New York: John Wiley & Sons, 2007.

[7] J.-R. De Boer, V. Calmettes, J.-Y. Tourneret, and B. Lesot, "Performance improvement of integrated MEMS/GNSS systems," *Proc. of ION-GNSS*, 2008.

[8] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, December 1989.

[9] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. New York: Prentice Hall, 1999.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, October 1986.

[11] C. Goodall, Z. Syed, and N. El-Sheimy, "Improving INS/GPS navigation accuracy through compensation of kalman filter errors," *Proc. of IEEE Vehicle and Technology Conference*, 2006.